UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/063,958	05/29/2002	Akira Koseki	JP920010018US1	2376

877 7590 12/16/2005

IBM CORPORATION, T.J. WATSON RESEARCH CENTER
P.O. BOX 218
YORKTOWN HEIGHTS, NY 10598

EXAMINER	
KENDALL, CHUCK O	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 12/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/063,958	KOSEKI ET AL.	
	Examiner	Art Unit	
	Chuck O. Kendall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05/29/02* is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the application filed 09/23/2005.

2. Claims 1 – 17 were previously presented. Claims 3 and 5 have been amended and claims 1 – 17 are still pending.

Response to Amendment

3. Examiner has withdrawn the 112, 2nd paragraph rejection (regarding lack of antecedent basis) in the non-final rejection of 06/23/05 regarding claims 5, 7 and 11, and also withdraws the previous non-final rejection of 06/23/2005 as well.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

The claimed invention is directed to non-statutory subject matter.

5. Claims 1 – 7, does not specifically recite the practical application and merely recites the steps of allocating instructions.

For eligibility under 35 U.S.C. 101, claims have to meet certain guidelines to be considered statutory:

> 1) "USEFUL RESULT"
>
> For an invention to be "useful" it must satisfy the utility requirement of section 101. The USPTO's official interpretation of the utility requirement provides that the utility of an invention has to be (i) specific, (ii) substantial and (iii) credible. MPEP Sec. 2107 and Fisher, 421 F.3d at ___, 76

USPQ2d at 1230 (citing the Utility Guidelines with approval for interpretation of "specific" and "substantial"). In addition, when the examiner has reason to believe that the claim is not for a practical application that produces a useful result, the claim should be rejected, thus requiring the applicant to distinguish the claim from the three Sec. 101 judicial exceptions to patentable subject matter by specifically reciting in the claim the practical application. In such cases, statements in the specification describing a practical application may not be sufficient to satisfy the requirements for section 101 with respect to the claimed invention. Likewise, a claim that can be read so broadly as to include statutory and nonstatutory subject matter must be amended to limit the claim to a practical application. In other words, if the specification discloses a practical application of a Sec. 101 judicial exception, but the claim is broader than the disclosure such that it does not require a practical application, then the claim must be rejected.

6. Regarding claims 8 – 10, and 12 – 14 the claimed invention is directed to non-statutory subject matter. Applicant, claims a compiler/software which is functionally descriptive embodied in a program, and does not include any hardware only software perse, hence the claim is neither concrete nor tangible, as well as also not specifically reciting any practical application.

For eligibility under 35 U.S.C. 101, claims have to meet certain guidelines to be considered statutory:

1) "USEFUL RESULT"

For an invention to be "useful" it must satisfy the utility requirement of section 101. The USPTO's official interpretation of the utility requirement provides that the utility of an invention has to be (i) specific, (ii) substantial and (iii) credible. MPEP Sec. 2107 and Fisher, 421 F.3d at ___, 76 USPQ2d at 1230 (citing the Utility Guidelines with approval for interpretation of "specific" and "substantial"). In addition, when the examiner has reason to believe that the claim is not for a practical application that produces a useful result, the claim should be rejected, thus requiring the applicant to distinguish the claim from the three Sec. 101 judicial exceptions to patentable subject matter by specifically reciting in the claim the practical application. In such cases, statements in the specification describing a practical application may not be sufficient to satisfy the

requirements for section 101 with respect to the claimed invention. Likewise, a claim that can be read so broadly as to include statutory and nonstatutory subject matter must be amended to limit the claim to a practical application. In other words, if the specification discloses a practical application of a Sec. 101 judicial exception, but the claim is broader than the disclosure such that it does not require a practical application, then the claim must be rejected.

2) "TANGIBLE"

Applying *In re Warmerdam*, 33 F.3d 1354, 31 USPQ2d 1754 (Fed. Cir. 1994), the examiner will determine whether there is simply a mathematical construct claimed, such as a disembodied data structure and method of making it. If so, the claim involves no more than a manipulation of an abstract idea and therefore, is nonstatutory under 35 U.S.C. § 101. In *Warmerdam* the **abstract idea** of a data structure became capable of producing a useful result when it was fixed in a tangible medium, which enabled its functionality to be realized.

3) "CONCRETE RESULT"

Another consideration is whether the invention produces a "concrete" result. Usually, this question arises when a result cannot be assured. In other words, the process must have a result that can be substantially repeatable or the process must substantially produce the same result again. In re Swartz, 232 F.3d 862,864, 56 USPQ2d 1703, 1704 (Fed. Cir. 2000) (where asserted result produced by the claimed invention is "irreproducible" claim should be rejected under section 101). The opposite of "concrete" is unrepeatable or unpredictable. Resolving this question is dependent on the level of skill in the art. For example, if the claimed invention is for a process which requires a particular skill, to determine whether that process is substantially repeatable will necessarily require a determination of the level of skill of the ordinary artisan in that field. An appropriate rejection under 35 U.S.C. Sec. 101 should be accompanied by a lack of enablement rejection under 35 U.S.C. Sec. 112, paragraph 1, where the invention cannot operate as intended without undue experimentation. See infra.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1 – 17 are rejected under 35 U.S.C. 102(b) as being anticipated by Wallace et al. USPN 6,018,799 (hereinafter "Wallace").

Regarding claim 1, Wallace anticipates a compiling method for converting into object code a program written in source code comprising the steps of:

allocating registers for a program to be compiled (4:1 – 5, see register stack and compile target programs); and

generating object code based on the register allocation, wherein said step of allocating registers includes the steps of allocating logical registers for instructions in said program (6:14 – 35, for logical registers see virtual registers, note: virtual registers are assigned to the pseudo registers as taught in 7:20 – 22, and are therefore both being interpreted as the logical registers), and performing mapping between said logical registers and physical registers so that said physical registers (6:22 – 25, see "virtual registers are assigned to the target computer's physical register) that are live at a procedure call (6:30 – 33, see physical registers and values of more immediate relevance) in said program to be compiled are allocated from the bottom of the register stack (13:55 – 65, see bottom-to-top).

Regarding claim 2, the compiling method according to claim 1, wherein, at said mapping step, allocation is done so that logical registers that are live across more procedure calls are first allocated (13:55 – 67, see "continues to a 'determine new live registers' procedure 803").

Regarding claim 3, the compiling method according to claim 1, wherein, at said mapping step, allocation is done, so that the logical registers that are allocated first are the logical registers that are live across a procedure call at which fewer logical registers are live at the same time (3:13 – 20, shows mapping one of the pseudo registers *logical registers,* to use the plurality of stack registers *physical registers* also, see FIG 5.A, 503, 505 and 507 and all associated text).

Regarding claim 4, Wallace anticipates a code generation method for generating code for a program that controls a computer comprising the steps of:

generating code while confirming that registers are allocated for a predetermined instruction (6:14 – 35, for logical registers see virtual registers); and

upon the calling of the procedure, so long as there is a vacancy in operation resources, copying said registers residing in the register stack, to free registers located at the bottom of said register stack (12:35 – 40, see saves copy of the register stack and 12:44 – 46, "707 that saves register stack state that exists at the end of the basic block").

Regarding claim 5, Wallace anticipates a method, for employing a stack register when a processor with a register stack executes a program, comprising the steps of:

when a different procedure is called in a predetermined procedure, said predetermined procedure called before the different procedure, reallocating registers that are allocated for the execution of said predetermined procedure and are live when said different procedure is called, and calling said different procedure (9:25 – 35, for call see "invoked", for live see "determines when the value in a register is dead"); and

upon the return from said different procedure, restoring the register image to the state immediately before the reallocation (12:35 – 38, shows copying register stack and 12: 47 – 50, shows normalizing the register stack, which Examiner interprets to be restoring register image).

Regarding claim 6, the stack register employment method according to claim 5, wherein said step of reallocating said registers and calling said different procedure includes the steps of:

sorting and reallocating, from the bottom of said register stack, said registers that are live when said different procedure is called (FIG. 5D, 557 also see all associated text, e.g. 13:57 – 67).

Regarding claim 7, Wallace anticipates a method, for employing a stack register when a program is executed by a processor with a register stack, comprising the steps of:

each time a procedure is called, packing and allocating existing logical registers (6:56 – 61, for *logical registers* see pseudo register and for *packing* see optimization);

performing said procedure, and restoring the register image to the state before the packing (12:35 – 38, shows copying register stack and 12: 47 – 50, shows normalizing the register stack, which Examiner interprets to be restoring register image).

Regarding claim 8, which claims similarly as already addressed limitations recited in claim 1, see rationale as previously discussed above.

Regarding claim 9, the compiler according to claim 8, wherein said register allocator allocates said logical registers and said physical registers first for an important portion of said program to be compiled (6:14 – 35, for logical registers see virtual registers, note: virtual registers are assigned to the pseudo registers as taught in 7:20 – 22, and are therefore both being interpreted as the logical registers); and

wherein, while for a less important portion of said program, said code generator generates compensation code for allocation of said logical registers and for allocation of said physical registers for the important portion (6:14 – 25, see code generator segment 211 and also see 6:55 – 61 for register allocation).

Regarding claim 10, Wallace anticipates a compiler for converting into machine language code the source code of a program written in a program language, the complier comprising:

a register allocator, for allocating registers for instructions in said program to be compiled (4:1 – 5, see register stack and compile target programs); and

a code generator, for generating object code based on the register allocation process performed by said register allocator, wherein said code generator generates code while confirming that registers are allocated for predetermined instructions (6:14 – 25, see code generator segment 211 and also see 6:55 – 61 for register allocation), and wherein, upon a procedure being called, said code generator, so long as there is a vacancy in operation resources, copies said registers residing in a register stack, to free registers that are located at the bottom of said register stack (6:30 – 33, see physical registers and values of more immediate relevance and 13:55 – 65, see determining new live register, also see 12:35 – 40, for see saves copy of the register stack and 12:44 – 46, "707 that saves register stack state that exists at the end of the basic block").

Regarding claim 11,Wallace anticipates a computer comprising:

input means, for entering source code of a program (5:33 – 35, see CD-ROM); and

a compiler, for compiling said source code and converting the compiled code into machine language code, wherein, before a different procedure is called in a predetermined procedure of a program to be compiled, said compiler generates code for reallocating registers that are allocated for the execution of said predetermined procedure and that are live when said different procedure is called (9:25 – 35, for called see "invoked", for live see "determines when the value in a register is dead"), and generates code, for restoring the register image, upon the return from said different

procedure, to the state immediately before the reallocation (12:35 – 38, shows copying register stack and 12: 47 – 50, shows normalizing the register stack, which Examiner interprets to be restoring register image).

Regarding claim 12, Wallace anticipates conversion program, for controlling a computer for conversion of a program to be executed, which permits said computer to perform, the conversion program comprising:

a process for allocating logical registers for instructions in said program to be executed (4:1 – 5, see register stack and compile target programs, also see 3:70 – 10 see pseudo registers for logical registers);

a process for performing mapping between said logical registers and physical registers, so that said physical registers that are live at a procedure call in said program to be compiled are allocated from the bottom of the register stack (3:29 – 33, see pseudo registers for *logical registers* and stack registers for *physical registers*); and

a process for generating object code based on the mapping process (6:14 – 25, see code generator segment 211 and also see 6:55 – 61 for register allocation).

Regarding claim 13, Wallace anticipates a conversion program, for controlling a computer for conversion of a program to be executed, which permits said computer to perform, the conversion program comprising:

a process for generating code while confirming that registers are allocated for a predetermined instruction (6:14 – 30, see code generator segment 211 and also see 6:55 – 61 for register allocation);

a process for, upon the calling of the procedure, so long as there is a vacancy in operation resources, copying said registers residing in said register stack, to free registers located at the bottom of said register stack (6:30 – 33, see physical registers and values of more immediate relevance and 13:55 – 65, see determining new live register, also see 12:35 – 40, for see saves copy of the register stack and 12:44 – 46, "707 that saves register stack state that exists at the end of the basic block").

Regarding claim 14, which describes the process version of claim 5, see rational as previously discussed above and with regards to a process see (FIG. 5A and 5B).

Regarding claim 15, Wallace anticipates a storage medium on which a conversion program is stored that controls a computer for conversion of a program to be executed, said conversion program permitting said computer to perform, the conversion program comprising:

a process for allocating logical registers for instructions in said program to be executed (4:1 – 5, see register stack and compile target programs);

a process for performing mapping between said logical registers and physical registers, so that said physical registers that are live at a procedure call in said program to be compiled are allocated from the bottom of the register stack (3:13 – 20, shows

mapping one of the pseudo registers *logical registers,* to use the plurality of stack registers *physical registers* also, see FIG 5.A, 503, 505 and 507 and all associated text*)*; and

a process for generating object code based on the mapping process(6:14 – 30, see code generator segment 211 and also see 6:55 – 61 for register allocation).

Regarding claim 16, Wallace anticipates a storage medium on which a conversion program is stored that controls a computer for conversion of a program to be executed, said conversion program permitting said computer to perform, the conversion program comprising:

a process for generating code while confirming that registers are allocated for a predetermined instruction(6:14 – 30, see code generator segment 211 and also see 6:55 – 61 for register allocation); and

a process for, upon the calling of the procedure, so long as there is a vacancy in operation resources, copying said registers residing in said register stack (12:35 – 40, see saves copy of the register stack and 12:44 – 46, "707 that saves register stack state that exists at the end of the basic block");

to free registers located at the bottom of said register stack (3:55 – 65, see determining new live register).

Regarding claim 17, Wallace anticipates a storage medium on which a conversion program is stored that controls a computer for conversion of a program to be

executed, said conversion program permitting said computer to perform, the conversion program comprising:

a process for, when a different procedure is called in a predetermined procedure, reallocating registers that are allocated for the execution of said predetermined procedure and that are live when said different procedure is called, and calling said different procedure (9:25 – 35, for call see "invoked", for live see "determines when the value in a register is dead"); and

a process for, upon the return from said different procedure, restoring the register image to the state immediately before the reallocation (12:35 – 38, shows copying register stack and 12: 47 – 50, shows normalizing the register stack, which Examiner interprets to be restoring register image).

Response to Arguments

9. Applicant's arguments with respect to claims 1 – 17 have been considered but are moot in view of the new ground(s) of rejection.

Correspondence information

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chuck Kendall whose telephone number is 571-272-3698. The examiner can normally be reached on 10:00 am - 6:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Ck.

Chuck Kendall Patent Examiner 2192
12/11/05